1   What is claimed is:

2

3   1. A method for forming an adjacent symbol codeword comprising:

4   generating a set of m bits, wherein m is an integer, of a first symbol and a set

5   of m bits of a second symbol from a first set of data during a first clock

6   phase; and

7   generating a set of n bits, wherein n is an integer, of the first symbol and a set

8   of n bits of the second symbol from a second set of data during a second

9   clock phase.

10

11   2. The method of claim 1 wherein the first and second set of data is from a memory.

12

13   3. The method of claim 2 wherein the memory is a Double Data Rate (DDR) memory.

14

15   4. The method of claim 1 wherein both m and n are four bits and constitute a nibble.

16

17   5. The method of claim 1 wherein the adjacent symbol codeword comprises an adjacent

18   formation of the first and second symbol.

19

20   6. The method of claim 1 further comprising isolating a common mode error across the

21   m and n bits of the first and second symbol.

22

23

24 7. A method for testing a memory comprising:

25   generating a plurality of check bits to append to data that is forwarded to the

26   memory;

27   generating an adjacent symbol codeword based at least in part on data received

28   from the memory;

29   decoding the adjacent symbol codeword; and

30   determining whether an error exists in the memory

31

32

33 8. The method of claim 7 wherein decoding the adjacent symbol codeword comprises

34 generating a syndrome based at least in part on the adjacent symbol codeword.

35

36 9. The method of claim 7 wherein determining whether an error exists in the memory is

37 based at least in part on the syndrome.

38

39 10. The method of claim 9 wherein an error exists based on the syndrome, further

40 comprising:

41   classifying the error in the received data; and

42   correcting the error in the received data.

43

44   11. The method of claim 7 wherein the memory is a Double Data Rate (DDR)

45   memory.

46

47   12. The method of claim wherein the syndrome is thirty two bits based on a two hundred

48   eighty eight bit codeword.

49

50   13. An apparatus for an Error Correcting Code comprising:

51   a first logic to generate a plurality of check bits based on a set of data, to append the

52   check bits to the set of data that is to be forwarded to a memory;

53   a second logic to receive a codeword from the memory and to generate a syndrome

54   based on the codeword, and to detect whether an error exists based on the syndrome;

55   a third logic to classify the error if it exists; and

56   a fourth logic to correct the error if it exists.

57

58   14. The apparatus of claim 13 is   incorporated within a server chipset.

59

60   15. The apparatus of claim 13 wherein the memory is a Double Data Rate (DDR)

61   memory.

62   16. The apparatus of claim 13 wherein the syndrome is thirty two bits and the codeword

63   is 288 bits.

64

65   17. The apparatus of claim 13 wherein the first logic is an encoder and utilizes the

66   formula:

67        $c_i = \Sigma\, d_j \times G_{ij}$ for i=0 to 31 and j= 0 to 255,

68   to generate the plurality of check bits.

69

70

71   18. The apparatus of claim 13 wherein the second logic is a decoder and the syndrome

72   is an H matrix that is generated by the formula:

73    $s_i = \Sigma \ v_j \ x \ H_{ij}$ for i=0 to 31 and j= 0 to 287,

74   to generate the syndrome.

75

76   19.   An apparatus to classify an error from a memory comprising:

77   a first logic to generate an H matrix syndrome; and

78   to determine whether an error exists based on the syndrome, if so, to classify an

79   error type of the error.

80

81   20. The apparatus of claim 19, wherein the H matrix syndrome is generated by the

82   formula:

83    $s_i = \Sigma \ v_j \ x \ H_{ij}$ for i=0 to 31 and j= 0 to 287.

84

85   21. The apparatus of claim 19 wherein to classify the error type comprises the first

86   logic to generate an error location vector.

87

88   22. The apparatus of claim 21 wherein the error location vector determines whether

89   the error is correctable:

90   a value of zero in the error location vector indicates the error is uncorrectable; in

91   contrast, a value that is greater than zero in an indicated column of the error location

92   vector indicates the error is correctable.

93   23. The apparatus of claim 19 wherein the error type may be either a single device

94   error or a double device error.

95

96   24. The apparatus of claim 23 wherein the double device error is either of a

97   simultaneous error or a sequential error.

98

99   25. The apparatus of claim 23 wherein the single device error is classified based on a

100   weight of an error value, e_0 and e_1, and the error location vector is to gate the error

101   values.

102

103

104   26. A system comprising:

105   a processor, coupled to a memory and a chipset, to generate an operation to

106   the memory via the chipset; and

107   the chipset to utilize an Error Correcting Code (ECC) based on an adjacent

108   symbol codeword that is formed in two clock phases and to determine

109   whether an error exists in a plurality of data received by the chipset from

110   the memory, if so, to classify a type of the error based on an H matrix

111   syndrome.

112

113   27. The system of claim 26 wherein the memory is

114   a Double Data Rate (DDR) memory.

115

116

117   28. The system of claim 26 wherein the H matrix syndrome is generated by the formula:

118     $s_i = \Sigma\ v_j\ x\ H_{ij}$ for i=0 to 31 and j= 0 to 287.

119

120

121   29. The system of claim 26 wherein the system is a server.